

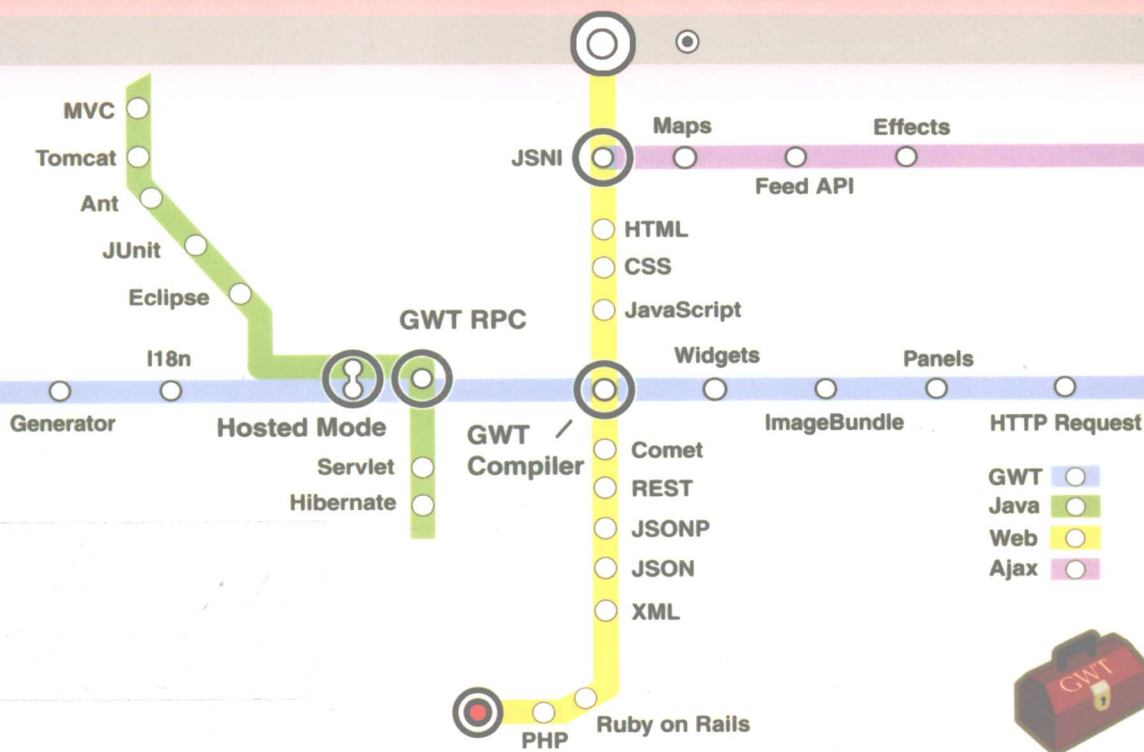


Google 核心技术丛书

# Google Web Toolkit 应用程序开发

# Google Web Toolkit Applications

(美) Ryan Dewsbury 著  
秦绪文 李松峰 译



机械工业出版社  
China Machine Press

7P393.09  
105

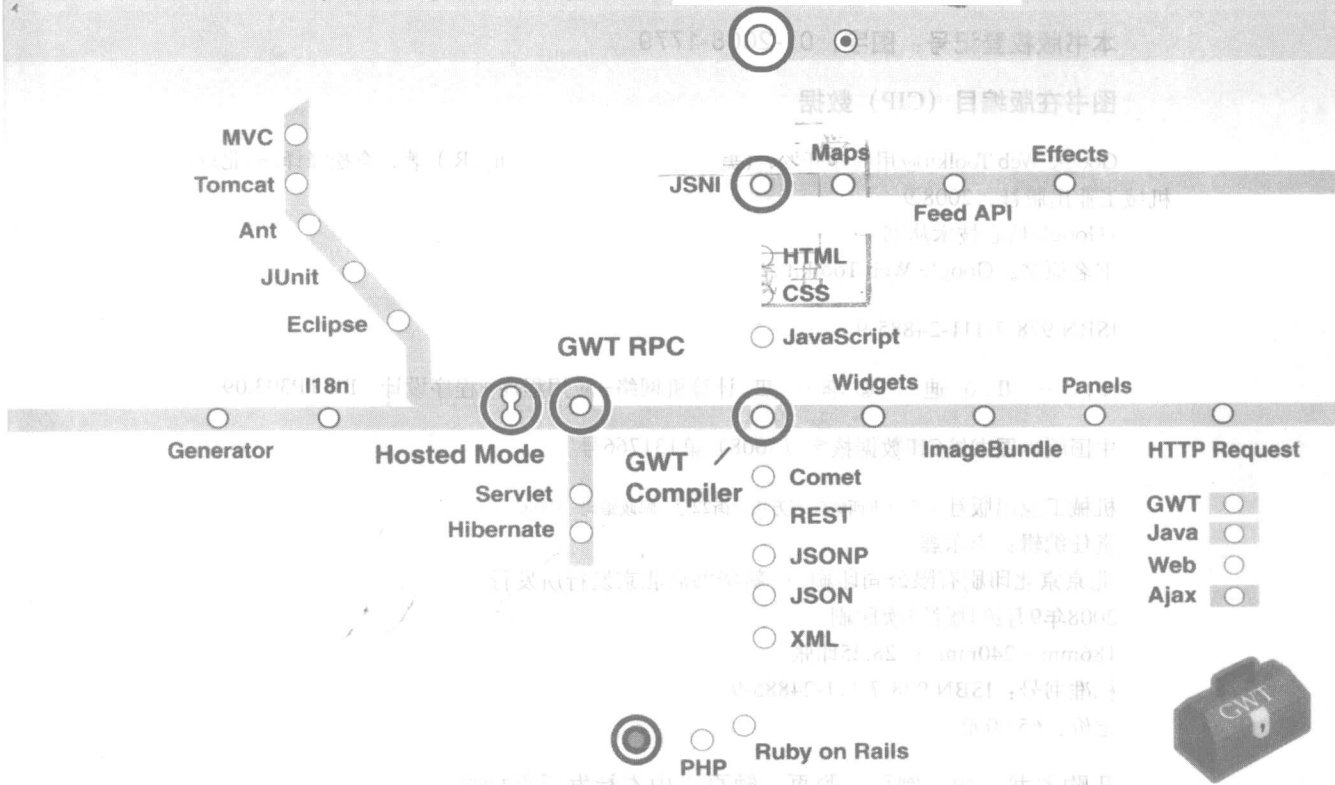
Google

# Google Web Toolkit

## 应用程序开发

### Google Web Toolkit Applications

(美) Ryan Dewsbury 著  
秦绪文 李松峰 译



机械工业出版社  
China Machine Press

本书为Web开发者使用GWT创建富Web应用程序,并向用户交付非同凡响的用户体验提供了必要的核心知识及使用指南。在这本实用性极强的书中,Ryan Dewsbury从权威人士的独特视角出发,以全面翔实的GWT技术阐释为基础,向读者详细解释了如何使用可靠的工具,创建一流的Web应用程序。

全书共10章,包括Google Web Toolkit简介、用户界面库、服务器整合技术、Ajax应用程序开发中的软件工程、高效使用GWT、Gadget desktop应用程序、Multi-Search应用程序、Blog Editor应用程序、Instant Messenger应用程序以及Database Editor应用程序等内容。

Simplified Chinese edition copyright © 2008 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Google web toolkit applications* (ISBN 978-0-321-50196-7) by Ryan Dewsbury, Copyright © 2008.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Sun Microsystems, Inc..

本书封面贴有Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2008-1779

### 图书在版编目(CIP)数据

Google Web Toolkit应用程序开发/(美)迪斯伯里(Dewsbury,R.)著;李松峰译.—北京:机械工业出版社,2008.9

(Google核心技术丛书)

书名原文:Google Web Toolkit Applications

ISBN 978-7-111-24885-9

I. G… II. ①迪… ②李… III. 计算机网络—应用程序—程序设计 IV. TP393.09

中国版本图书馆CIP数据核字(2008)第131766号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:李东震

北京京北印刷有限公司印刷·新华书店北京发行所发行

2008年9月第1版第1次印刷

186mm×240mm·28.25印张

标准书号:ISBN 978-7-111-24885-9

定价:65.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换  
本社购书热线(010)68326294

## 本书赢得的赞誉

“Ryan清楚地知道GWT的价值所在，更对GWT与多种Web技术的整合了若指掌——而且，不仅仅是理论上的了解。随着gpokr.com和kdice.com的声名鹊起，Ryan已经成为业界名符其实的GWT权威人物。”

——Bruce Johnson, Google Web Toolkit创建者

“本书区别于其他GWT图书的一个重要特点，就是它以构建经过为线索，全程剖析了一些非同凡响的GWT应用程序，而非多数图书中展示的玩具式应用程序。”

——R. Mark Volkman, Object Computing, Inc. 合伙人

“对任何使用GWT做开发的人来说，本书都堪称卓越。它围绕真实应用程序的设计与开发，对使用GWT时所遇到的种种挑战都一一给出解决之道。全书介绍的应用程序充分展示了开发过程中的最佳实践，从简单的UI设计直至自定义代码生成，而理解这些的一个小前提就是开发者必须具备必要的Java知识。当然，本书也没有回避对各种高级概念的解释，而且其解释方式无论对于新手还是经验丰富的开发者，都非常容易接受。良好的应用程序开发实践以及适当的模型视图控制器设计，始终贯穿全书，为本书的读者在开发实践中更上一层楼提供了切实保障。”

——Jason Essington, Green River Computing Services Inc. 高级Web/Java工程师

“Dewsbury的这本书既面向专家，也适合想要使用GWT这个开源Java软件开发框架编写Ajax应用程序的初学者。这是一本内容翔实、深入浅出的好书！”

——Massimo Nardone, Advisory IT Security架构师, IBM芬兰Finnish Invention Development Team Leader

## 前言

我始终保持着对软件开发非技术一面——用户体验的兴趣。说到这个兴趣的由来，大概要追溯到当年我和一个团队用C++来构建某个应用程序服务器的核心。当时，大家对C++语言的优美及其可表达性都很欣赏。我们以优雅的代码完成了几个大型的复杂系统，它们都能够流畅运行。其中，最令人称奇的大概要数我们使用的模板技术，该技术能够使C++编译器像代码生成器一样，帮我们生成烦琐的代码。后来，每当跟别人提起那种美妙的用户体验时，我看到的往往都是一些茫然的眼神。

我决定抽时间写一个客户端应用程序，它带给用户的优雅体验，就如同开发者看到写得漂亮的代码一般。我选择了构建一个即时消息服务，主要通过C++来开发，这个应用程序将4种主要的网络组合到了一个界面中。那时候，即时消息软件的功能日益膨胀——众多按钮令只想发送一条简单的文本消息的用户感到无所适从。而我所开发的这个即时消息程序则能够给用户带来更好的体验：用户不必下载10MB的客户端，也无需经历5步的安装过程，经过我的优化，这个程序只有200KB，而且界面整洁（很像我们今天使用的Google Talk）。结果，这个程序获得了百万级的下载量。

在使用C++开发界面的过程中，创建漂亮的网页界面是那样简单，而我总是对此印象深刻。其实，只要将C++与层叠样式表中设置字体的代码作一比较，你就会明白我的意思。随后，Ajax开始流行，而通过Ajax也能构建类似桌面应用程序的Web应用。加之使用CSS能轻易得到漂亮的外观，因此Ajax就成为人们更好的界面开发平台。

当我第一次看到Google Maps时，我确实被深深地触动了。那种用户体验简直太棒了！只要在浏览器中输入maps.google.com，马上就能使用一套全功能的地图应用程序。既可以在各个方向上拖动地图，也可以让它带着你周游世界，而且无论放大还是缩小都无需等待页面刷新。于是，我大概地看了一下实现这些功能所需的技术——尤其是JavaScript，结果非常令人失望。我知道使用JavaScript存在着很多的局限性。由于这些局限性，几乎不可能用它来构建一个大型复杂的客户端应用程序。

后来，Google Web Toolkit (GWT) 发布了。我决定用它来试着编写一个应用程序。只用了3周时间，我就构建完成了poker应用程序的客户端和服务端。我把它放到了<http://gpokr.com>上。只要在浏览器中加载这个URL，就可以立即启动一个实时的poker游戏。无需下载、无需安装，而且通过CSS能够为界面添加漂亮的样式。GWT团队的Scott Blum、Bruce Johnson以及Joel Webber，都闻讯前来“测试”。因此，我也有了一个向他们道谢的机会，感谢他们开发了这

令人惊叹的工具。最让我激动不已的就是，GWT编译器能够把优美的Java代码转换成JavaScript。于是，任何人都可以创建向用户交付美好体验的应用程序，而可靠的GWT则给我留下了深刻的印象。

在GWT首次发布之后，我发现许多人还对它的强大功能一无所知。显然，最好能有一本书通过实例向大家展示GWT各方面的能力。此前，我没有写书的经验，而且写一本并非我专业的技术书，看起来似乎也不合适。不过，当时可以说也没有什么人对GWT很在行。出于对这一技术的信赖，我决定试一试。为了弥补经验的不足，在动笔写这本书之前，我先用几个月的时间专门开发了一些GWT应用程序。通过这些应用程序的开发，我探索了GWT中的各个部分，也搞清了GWT可能会涉及的所有Web技术。本书的第二部分展示了这些应用程序。

## 本书重点内容

本书讲解如何使用Web技术和Java开发工具，以GWT为桥梁，创建能为用户提供卓越体验的Ajax应用程序。本书的焦点主要是Google Web Toolkit，同时也深入介绍了它提供的库和工具。作为第二个焦点，本书也讨论了使用Java的软件开发技术和模式，以及如何使用GWT来开发Ajax应用程序。第三个焦点是Web技术，包括Web标准和其他Ajax库及API。

## 本书读者对象

作为一名开发者，我写这本书的目的，是为了给其他开发者看。如果你想要创建面向用户的应用程序，那么你应该看这本书。本书中的大部分代码都基于Java，但由于精心编排，本书也适合Java语言初学者阅读。如果你想看这本书，但又不懂Java，那么就需要在阅读本书之前先学习Java。Sun公司为学习Java的开发者提供了丰富的在线教程，在<http://java.sun.com/docs/books/tutorial/java/index.html>可以找到。

对于Java开发者而言，GWT不仅仅是一个Ajax工具。在我看来，这种观点严重低估了GWT的能力。虽然Java开发者使用它会感觉很简单；但是，只要想构建不同凡响的Ajax应用程序，那么无论你是.NET、PHP、Ruby开发者，还是C++开发者，这本书同样也适合你。假如你是上述开发者之一，想要掌握另一个门语言来构建Ajax应用程序——无论用不用GWT；那么，我的建议是学习Java——从前面提到的Sun公司的在线教程开始，并通过本书来学习GWT，以及学习<http://code.google.com/webtoolkit/documentation/>中的GWT文档，而不是学习JavaScript。这样，你就会节省大量时间用于测试和维护应用程序，同时还能创造出更好的用户体验。

## 本书结构

本书分为两部分。第一部分是対Google Web Toolkit的深入介绍。可以将这一部分作为

GWT库的参考，或者作为使用GWT中高效开发技术的指南。第二部分全面剖析了5个通过GWT构建的不同凡响的应用程序。这一部分将介绍应用程序设计与开发过程中涉及到的开发模式、方法和技巧。这一部分中的每个应用程序，都是对GWT库的应用、Web服务、技术搭配、应用程序设计和架构，以及用户界面设计等因素综合考量的结果。在阅读这些章节时，读者可以在自己的机器中同步构建应用程序。虽然每一章都包含大部分相关代码，但在讲解某些实例时，读者可能需要根据提示查看本书的源代码。本书的源代码可以在[www.gwtapps.com](http://www.gwtapps.com)下载到（打开页面后单击左侧栏中Book Forum链接，然后单击“文件”，再选择下载各章的代码或全部源代码）。

### 第一部分：理解Google Web Toolkit

第1章，Google Web Toolkit简介。讨论了Web技术、能力组合及GWT。最后展示了一个创建Ajax游戏应用程序的简短教程。

第2章，用户界面库。详细介绍了GWT提供的用户界面库，主要围绕每个部件的用法，由说明和例子构成。

第3章，服务器整合技术。讨论了与服务器端应用程序进行整合的几种方法。

第4章，Ajax应用程序开发中的软件工程。着眼于用于软件开发的Java工具，以及如何将这些工具应用到GWT开发中。

第5章，高效使用GWT。揭示了GWT开发中的一些高级技术，包括CSS、代码生成、国际化以及性能优化。

### 第二部分：创建富Web应用程序

第6章，Gadget Desktop应用程序。这是一个带有丰富界面拖放功能的桌面配件应用程序，涉及使用cookie和Gears实现持久化，以及在GWT中使用JavaScript API。

第7章，Multi-Search应用程序。展示了如何创建向多个搜索引擎发送请求的搜索应用程序。该应用程序使用参数式JavaScript对象表示法(JavaScript Object Notation with Padding, JSONP)，实现了与Google、Yahoo!、Amazon以及Flickr的通信。

第8章，Blog Editor应用程序。本章展示了一个用于管理多个博客中的文章的应用程序。该应用程序通过HTTP代理整合了Blogger REST API。

第9章，Instant Messenger应用程序。详细剖析了基于GWT-RPC在网页中构建即时通信系统的过程。该应用程序还涉及到使用基于事件的协议，以及通过Tomcat中的Comet及Jetty中的Continuations实现优化。

第10章，Database Editor应用程序。关注的是一个在传统网页中实现数据库管理的应用程序。该应用程序涵盖了很多高级主题，例如：使用数据访问对象（Data Access Object, DAO）从服务器上读取复杂的数据结构、简化XML和JSON的代码生成，以及同PHP、Ruby on Rails及Java与Hibernate整合。

## 在线支持

本书的网站是[www.gwtapps.com](http://www.gwtapps.com)。这个网站中包含了本书的源代码及各个示例应用程序的实时演示版。单击页面左上方的Book Forum链接，可以进入一个论坛，供读者提问、报告错误和下载源代码。同时，网站中还提供了其他参考资料。

## 致谢

我感谢GWT团队，感谢他们开发了这样一项值得大书特书的技术。同样，我要感谢Prentice Hall出版社那些优秀的人士：John Wait，他在设计和创新方面总有各种奇思妙想；Rebecca Greenberg，她的编辑工作令人赞叹；Chris Zahn，他心细如发并且激情四射；Raina Chrobak，他做事极有条理并容忍了我对目录三番五次的修改，还有Julie Nahil，感谢他对本书制作所付出的努力。

感谢所有审稿人：Sandy McArthur、Jason Essington、Bruce Johnson、Massimo Nardone和Mark Volkmann。感谢他们发现的错误和提出的建议，减少了本书存在的问题。

最后，感谢Gpokr和Kdice的数以千计的玩家，他们容忍了我长达6个月没有升级程序和修复bug。

## 作者简介

作为一名开发者、架构师和咨询师，Ryan Dewsbury从1998年起就开始使用C++和Java。他曾花费几年时间协助完成了一个系统框架的构建工作，该框架设计用于一种半导体制造系统。最近，Ryan正致力于协助几家刚刚成立的Web公司，使用前沿软件来创造卓越的用户体验。

工作之余，Ryan还投入时间开发独立的软件项目，包括2004年完成的Easy Message，以及近期的Gpokr (<http://gpokr.com>) 和KDice (<http://kdice.com>)。后两者是他基于GWT在不经意间开发成功的在线游戏。

# 目 录

## 前言

## 第一部分 理解Google Web Toolkit

|                           |    |
|---------------------------|----|
| 第1章 Google Web Toolkit简介  | 1  |
| 1.1 Ajax崭露头角              | 1  |
| 1.2 重新定义Web应用程序           | 2  |
| 1.2.1 Adobe的Flash与Flex    | 3  |
| 1.2.2 微软的Silverlight      | 4  |
| 1.2.3 Java FX             | 4  |
| 1.3 基于Ajax的软件工程           | 4  |
| 1.3.1 通过部件和面板构建丰富的用户界面    | 5  |
| 1.3.2 利用异步通信获得更好的性能       | 6  |
| 1.3.3 通过Web标准和Web服务提供互操作性 | 6  |
| 1.3.4 使用Java工具提高开发效率      | 6  |
| 1.4 职业背景评估                | 7  |
| 1.4.1 Web设计者              | 7  |
| 1.4.2 网站开发者               | 7  |
| 1.4.3 Ajax开发者             | 8  |
| 1.4.4 Web应用程序开发者          | 8  |
| 1.4.5 桌面应用程序开发者           | 8  |
| 1.5 应用程序开发技能的重要性          | 8  |
| 1.6 简单的教程                 | 10 |
| 1.6.1 创建GWT项目             | 10 |
| 1.6.2 自定义生成的应用程序          | 13 |
| 1.6.3 创建动态的界面             | 15 |
| 1.6.4 使用异步HTTP            | 20 |
| 1.7 GWT中的包                | 22 |
| 1.8 GWT应用程序概述             | 25 |
| 1.8.1 常见的应用程序模式           | 25 |
| 1.8.2 示例应用程序简介            | 25 |

|                                               |    |
|-----------------------------------------------|----|
| 1.9 小结                                        | 26 |
| 第2章 用户界面库                                     | 27 |
| 2.1 静态部件                                      | 28 |
| 2.1.1 Label                                   | 28 |
| 2.1.2 HTML                                    | 30 |
| 2.1.3 Image                                   | 31 |
| 2.1.4 Hyperlink                               | 32 |
| 2.2 表单部件                                      | 34 |
| 2.2.1 Button                                  | 35 |
| 2.2.2 ToggleButton和PushButton                 | 36 |
| 2.2.3 Checkbox                                | 39 |
| 2.2.4 RadioButton                             | 39 |
| 2.2.5 ListBox                                 | 40 |
| 2.2.6 SuggestBox                              | 41 |
| 2.2.7 TextBox                                 | 43 |
| 2.2.8 PasswordTextBox                         | 44 |
| 2.2.9 TextArea                                | 44 |
| 2.2.10 RichTextArea                           | 45 |
| 2.2.11 FileUpload                             | 46 |
| 2.2.12 Hidden                                 | 46 |
| 2.3 复杂部件                                      | 46 |
| 2.3.1 Tree                                    | 47 |
| 2.3.2 MenuBar                                 | 50 |
| 2.4 简单布局面板                                    | 52 |
| 2.4.1 FlowPanel                               | 52 |
| 2.4.2 HorizontalPanel和VerticalPanel           | 53 |
| 2.4.3 HorizontalSplitPanel和VerticalSplitPanel | 54 |
| 2.4.4 FlexTable和Grid                          | 55 |
| 2.4.5 DeckPanel                               | 58 |
| 2.4.6 DockPanel                               | 59 |
| 2.4.7 HTMLPanel                               | 60 |

|                              |     |                                  |     |
|------------------------------|-----|----------------------------------|-----|
| 2.5 复杂布局面板 .....             | 61  | 4.1.1 安装JDK .....                | 105 |
| 2.5.1 StackPanel .....       | 61  | 4.1.2 安装GWT .....                | 105 |
| 2.5.2 TabPanel .....         | 62  | 4.1.3 安装Eclipse .....            | 108 |
| 2.6 简单容器面板 .....             | 64  | 4.2 向Eclipse中添加项目 .....          | 109 |
| 2.6.1 Composite .....        | 64  | 4.3 在Eclipse中编写Java代码 .....      | 112 |
| 2.6.2 SimplePanel .....      | 65  | 4.3.1 在Eclipse中创建类 .....         | 113 |
| 2.6.3 ScrollPanel .....      | 65  | 4.3.2 使用Eclipse的Java编辑器 .....    | 116 |
| 2.6.4 FocusPanel .....       | 65  | 4.4 在Eclipse中进行调试 .....          | 119 |
| 2.7 复杂容器面板 .....             | 66  | 4.5 组织应用程序的结构 .....              | 124 |
| 2.7.1 FormPanel .....        | 66  | 4.6 测试应用程序 .....                 | 127 |
| 2.7.2 DisclosurePanel .....  | 68  | 4.6.1 使用JUnit .....              | 129 |
| 2.7.3 PopupPanel .....       | 69  | 4.6.2 基准测试 .....                 | 133 |
| 2.7.4 DialogBox .....        | 70  | 4.7 构建和共享模块 .....                | 136 |
| 2.8 用户界面框架 .....             | 73  | 4.7.1 使用模块 .....                 | 136 |
| 2.8.1 事件接口 .....             | 73  | 4.7.2 创建可重用的模块 .....             | 138 |
| 2.8.2 特性接口 .....             | 76  | 4.7.3 共享编译后的应用程序 (Mashup) .....  | 139 |
| 2.9 小结 .....                 | 78  | 4.8 部署应用程序 .....                 | 139 |
| 第3章 服务器整合技术 .....            | 79  | 4.8.1 部署到Web服务器 .....            | 139 |
| 3.1 基本的服务器整合技术 .....         | 79  | 4.8.2 向Servlet容器中部署Servlet ..... | 141 |
| 3.1.1 异步HTTP请求 .....         | 79  | 4.8.3 使用Ant自动完成部署 .....          | 142 |
| 3.1.2 操作纯文本和HTML .....       | 81  | 4.9 小结 .....                     | 145 |
| 3.1.3 整合传统的服务器端脚本技术 .....    | 83  | 第5章 高效使用GWT .....                | 146 |
| 3.2 使用数据格式库 .....            | 87  | 5.1 深入理解异步编程 .....               | 146 |
| 3.2.1 读写XML数据 .....          | 88  | 5.2 处理后退按钮 .....                 | 149 |
| 3.2.2 读写JSON数据 .....         | 92  | 5.3 使用CSS创建美观的界面 .....           | 152 |
| 3.3 第三方服务器整合 .....           | 94  | 5.3.1 将GWT部件与CSS关联起来 .....       | 152 |
| 3.3.1 同源策略 .....             | 95  | 5.3.2 指定颜色 .....                 | 154 |
| 3.3.2 使用JavaScript API ..... | 95  | 5.3.3 指定单位 .....                 | 155 |
| 3.3.3 使用JSONP .....          | 96  | 5.3.4 字体属性 .....                 | 156 |
| 3.3.4 使用HTTP代理 .....         | 97  | 5.3.5 文本属性 .....                 | 157 |
| 3.4 高级服务器整合技术 .....          | 97  | 5.3.6 外边距、边框和内边距属性 .....         | 157 |
| 3.4.1 有状态服务器 .....           | 97  | 5.3.7 CSS示例 .....                | 159 |
| 3.4.2 整合无状态服务器 .....         | 98  | 5.3.8 光标属性 .....                 | 160 |
| 3.4.3 使用GWT-RPC .....        | 100 | 5.4 扩展GWT .....                  | 162 |
| 3.5 小结 .....                 | 104 | 5.4.1 构建新部件 .....                | 162 |
| 第4章 Ajax应用程序开发中的软件工程 .....   | 105 | 5.4.2 使用JSNI .....               | 164 |
| 4.1 设置开发环境 .....             | 105 | 5.4.3 提供浏览器支持 .....              | 166 |

|                                           |     |                                                 |     |
|-------------------------------------------|-----|-------------------------------------------------|-----|
| 5.4.4 使用其他JavaScript库                     | 169 | 6.4.2 编写配件容器                                    | 209 |
| 5.5 应用程序国际化                               | 173 | 6.4.3 使用ImageBundle、ToggleButton<br>和PushButton | 213 |
| 5.5.1 声明常量                                | 173 | 6.5 创建拖放部件                                      | 215 |
| 5.5.2 声明消息                                | 174 | 6.5.1 在面板之间停放部件                                 | 216 |
| 5.5.3 本地化工具                               | 176 | 6.5.2 通过鼠标事件拖动部件                                | 217 |
| 5.5.4 本地化管理                               | 178 | 6.5.3 停放部件                                      | 220 |
| 5.5.5 使用字典                                | 179 | 6.6 通过CSS设计用户界面                                 | 221 |
| 5.6 生成代码                                  | 179 | 6.6.1 通过GWT使用CSS                                | 222 |
| 5.6.1 使用生成的代码                             | 179 | 6.6.2 为选项卡添加样式                                  | 224 |
| 5.6.2 编写代码生成器                             | 181 | 6.6.3 为菜单添加样式                                   | 226 |
| 5.6.3 generate方法分析                        | 182 | 6.6.4 为GadgetContainerView添加样式                  | 226 |
| 5.7 性能优化                                  | 185 | 6.6.5 创建RoundedPanel                            | 228 |
| 5.7.1 处理耗时过程                              | 185 | 6.7 增加持久性                                       | 229 |
| 5.7.2 使用ImageBundle                       | 186 | 6.7.1 实现CookieStorageClass类                     | 229 |
| 5.7.3 在Apache中设置缓存                        | 188 | 6.7.2 处理浏览器差异                                   | 231 |
| 5.7.4 在Servlet容器中设置缓存                     | 189 | 6.7.3 加载和保存cookie                               | 232 |
| 5.7.5 在Apache中设置压缩                        | 190 | 6.7.4 使用Google Gears存储信息                        | 233 |
| 5.7.6 在Servlet容器中设置压缩                     | 190 | 6.8 使用其他JavaScript库                             | 237 |
| 5.8 小结                                    | 192 | 6.8.1 使用Google Maps API                         | 237 |
| <b>第二部分 创建富Web应用程序</b>                    |     | 6.8.2 使用Google Ajax API加载新闻源                    | 239 |
| <b>第6章 Gadget Desktop应用程序</b>             | 193 | 6.8.3 构建天气预报配件                                  | 241 |
| 6.1 使用容器应用程序模式                            | 193 | 6.8.4 构建RSS新闻配件                                 | 243 |
| 6.2 模型设计                                  | 194 | 6.9 小结                                          | 244 |
| 6.2.1 使用抽象工厂模式                            | 195 | <b>第7章 Multi-Search应用程序</b>                     | 246 |
| 6.2.2 编写GadgetClass类                      | 195 | 7.1 使用聚合器应用程序模式                                 | 246 |
| 6.2.3 编写Gadget类                           | 197 | 7.2 Multi-Search应用程序设计                          | 247 |
| 6.2.4 体现用户偏好                              | 198 | 7.3 模型                                          | 248 |
| 6.3 构建分栏的容器界面                             | 200 | 7.4 视图                                          | 249 |
| 6.3.1 定义视图                                | 201 | 7.4.1 MultiSearchView类                          | 250 |
| 6.3.2 使用VerticalPanel构建主布局                | 202 | 7.4.2 SearchResultsView类                        | 253 |
| 6.3.3 使用HorizontalPanel构建菜单               | 202 | 7.4.3 构建Link部件                                  | 254 |
| 6.3.4 使用TabPanel构建页面选择区                   | 203 | 7.5 控制器                                         | 255 |
| 6.3.5 使用HorizontalPanel和FlowPanel<br>构建分栏 | 205 | 7.6 导入结构化的数据格式                                  | 257 |
| 6.4 将部件放入视图                               | 208 | 7.6.1 绕过同源策略的限制                                 | 258 |
| 6.4.1 Hello World配件                       | 208 | 7.6.2 通过JSONP加载JSON源                            | 259 |
|                                           |     | 7.7 与Yahoo! Search进行交互                          | 262 |

|                                                   |     |
|---------------------------------------------------|-----|
| 7.8 整合Google Base .....                           | 266 |
| 7.9 整合Flickr Search .....                         | 269 |
| 7.10 整合Amazon Search .....                        | 274 |
| 7.11 小结 .....                                     | 277 |
| 第8章 Blog Editor应用程序 .....                         | 279 |
| 8.1 使用工作区应用程序模式 .....                             | 279 |
| 8.2 构建Web服务客户端 .....                              | 280 |
| 8.3 Blog Editor应用程序设计 .....                       | 282 |
| 8.4 模型 .....                                      | 282 |
| 8.5 构建多文档视图 .....                                 | 284 |
| 8.5.1 BlogEditorView类 .....                       | 285 |
| 8.5.2 BlogView类 .....                             | 287 |
| 8.5.3 BlogEntryView类 .....                        | 291 |
| 8.5.4 EditEntryView类 .....                        | 293 |
| 8.6 添加富文本编辑功能 .....                               | 296 |
| 8.6.1 使用RichTextArea .....                        | 296 |
| 8.6.2 使用ImageBundle、国际化和<br>RichTextToolbar ..... | 297 |
| 8.7 LoadingPanel部件 .....                          | 299 |
| 8.8 TitleCommandBar部件 .....                       | 301 |
| 8.9 设计应用程序的控制器 .....                              | 303 |
| 8.10 构建作为HTTP代理的Servlet .....                     | 304 |
| 8.10.1 新的跨域的RequestBuilder .....                  | 305 |
| 8.10.2 编写代理Servlet .....                          | 308 |
| 8.11 与Blogger API整合 .....                         | 312 |
| 8.11.1 使用Atom Publishing Protocol和<br>GData ..... | 312 |
| 8.11.2 定义BloggerService类 .....                    | 313 |
| 8.11.3 登录到Google账户 .....                          | 314 |
| 8.11.4 从账户中获取XML格式的博主<br>列表 .....                 | 319 |
| 8.11.5 获取每个博客中XML格式的文章<br>列表 .....                | 322 |
| 8.11.6 发送XML创建及保存文章 .....                         | 324 |
| 8.11.7 发送删除一篇文章的Delete请求 .....                    | 327 |
| 8.12 小结 .....                                     | 328 |

|                                     |     |
|-------------------------------------|-----|
| 第9章 Instant Messenger应用程序 .....     | 330 |
| 9.1 使用合作者应用程序模式 .....               | 330 |
| 9.2 Instant Messenger应用程序设计 .....   | 331 |
| 9.3 模型 .....                        | 332 |
| 9.4 构建补充性界面 .....                   | 334 |
| 9.4.1 MessengerView类 .....          | 336 |
| 9.4.2 SignInView类 .....             | 340 |
| 9.4.3 ContactListView类 .....        | 342 |
| 9.4.4 ChatWindowView类 .....         | 345 |
| 9.5 控制器概述 .....                     | 350 |
| 9.6 使用GWT-RPC .....                 | 352 |
| 9.6.1 MessengerService接口 .....      | 353 |
| 9.6.2 RemoteServiceServlet类 .....   | 354 |
| 9.6.3 使用异步接口 .....                  | 355 |
| 9.7 连接到服务器 .....                    | 356 |
| 9.8 添加RPC事件 .....                   | 358 |
| 9.8.1 轮询协议 .....                    | 358 |
| 9.8.2 基于事件的协议 .....                 | 359 |
| 9.8.3 实现事件 .....                    | 361 |
| 9.9 Instant Messenger的服务器 .....     | 363 |
| 9.10 使用服务器端高级IO .....               | 368 |
| 9.10.1 在Tomcat中使用Comet .....        | 370 |
| 9.10.2 在Jetty中使用Continuations ..... | 373 |
| 9.11 小结 .....                       | 375 |
| 第10章 Database Editor应用程序 .....      | 376 |
| 10.1 使用管理员应用程序模式 .....              | 376 |
| 10.2 模型设计 .....                     | 378 |
| 10.3 使用异步数据访问对象 .....               | 381 |
| 10.4 构建双栏编辑器界面 .....                | 383 |
| 10.4.1 使用Tree和SplitPanel部件 .....    | 383 |
| 10.4.2 扩展及动态加载树形项 .....             | 385 |
| 10.4.3 创建工作区视图 .....                | 388 |
| 10.4.4 使用对话框实现编辑和创建对象 .....         | 396 |
| 10.5 服务器整合概览 .....                  | 399 |
| 10.5.1 使用Action .....               | 400 |
| 10.5.2 使用REST .....                 | 401 |

|                             |     |
|-----------------------------|-----|
| 10.5.3 使用RPC                | 401 |
| 10.6 编写通用的GWT代码生成器          | 402 |
| 10.6.1 编写代码生成器              | 404 |
| 10.6.2 自动串行化为XML            | 411 |
| 10.6.3 自动串行化为JSON           | 411 |
| 10.7 整合基于Action的PHP脚本       | 411 |
| 10.7.1 使用PHP构建Action API    | 412 |
| 10.7.2 编写客户端的Action数据访问层    | 415 |
| 9.9.1 MessageService接口      | 333 |
| 9.9.2 RemoteServiceService类 | 334 |
| 9.9.3 使用异步接口                | 335 |
| 9.7 连接到服务器                  | 328 |
| 9.8 添加RPC事件                 | 328 |
| 9.8.1 拖拽特效                  | 328 |
| 9.8.2 基于事件的协作               | 329 |
| 9.8.3 实现事件                  | 331 |
| 9.9 Instant Messenger的服务器   | 333 |
| 9.10 使用服务器端流式IO             | 336 |
| 9.10.1 在Joomla中使用Comet      | 370 |
| 9.10.2 在Jelly中使用Comet       | 373 |
| 9.11 小结                     | 375 |
| 第10章 Database Editor应用程序    | 376 |
| 10.1 使用管理员应用程序模式            | 376 |
| 10.2 搭建设计                   | 378 |
| 10.3 使用异步数据源访问对象            | 381 |
| 10.4 构建双栏编辑器界面              | 383 |
| 10.4.1 使用Tree的Panel组件       | 383 |
| 10.4.2 使用及添加树状图             | 385 |
| 10.4.3 创建工作区视图              | 388 |
| 10.4.4 使用树结构实现编辑和创建对象       | 390 |
| 10.5 取消数据源连接                | 390 |
| 10.5.1 使用Action             | 400 |
| 10.5.2 使用REST               | 401 |

|                                   |     |
|-----------------------------------|-----|
| 10.8 整合REST风格的Ruby on Rails应用     | 418 |
| 程序                                | 419 |
| 10.8.1 使用Ruby on Rails构建REST API  | 419 |
| 10.8.2 编写客户端的REST数据访问层            | 425 |
| 10.9 整合GWT-RPC Servlet            | 428 |
| 10.9.1 编写RPC服务                    | 429 |
| 10.9.2 通过Hibernate存储模型            | 433 |
| 10.10 小结                          | 439 |
| 8.4 摘要                            | 385 |
| 8.2 构建多文档视图                       | 384 |
| 8.2.1 BlogEditorView类             | 385 |
| 8.2.2 BlogView类                   | 387 |
| 8.2.3 BlogEntryView类              | 391 |
| 8.2.4 EditEntryView类              | 393 |
| 8.6 添加富文本编辑功能                     | 396 |
| 8.6.1 使用RichTextArea              | 396 |
| 8.6.2 使用ImageBundles 国际化组         | 397 |
| RichTextToolbar                   | 397 |
| 8.7 LoadingPanel组件                | 399 |
| 8.8 TitleCommandBar组件             | 399 |
| 8.9 设计应用程序的控制器                    | 399 |
| 8.10 构建作为HTTP代理的Servlet           | 399 |
| 8.10.1 新的Servlet的RequestBuilder   | 399 |
| 8.10.2 编写代理Servlet                | 399 |
| 8.11 与Blogger API整合               | 399 |
| 8.11.1 使用Atom Publishing Protocol | 399 |
| CDATA                             | 399 |
| 8.11.2 定义BlogService类             | 399 |
| 8.11.3 登录到Google账户                | 399 |
| 8.11.4 从账户中获取XML格式的博客             | 399 |
| 列表                                | 399 |
| 8.11.2 获取每个博客中XML格式的文章            | 399 |
| 列表                                | 399 |
| 8.11.6 发布XML创建及保存文章               | 399 |
| 8.11.7 发布删除一篇文章的Delete请求          | 399 |
| 8.12 小结                           | 399 |

# 第一部分 理解Google Web Toolkit

## 第1章 Google Web Toolkit简介

本章通过回顾Web技术的发展来帮助读者理解GWT (Google Web Toolkit, Google Web Toolkit) 作为一种Web应用程序开发技术和工具的定位。同时, 还展示如何使用GWT构建简单的Ajax应用程序, 并简单地介绍GWT的库。

### 1.1 Ajax崭露头角

21世纪初, 随着.com泡沫的破灭和浏览器战争的平息, 许多公司都不再看好发展Web技术的前景, 从而导致原先预示着未来方向的许多技术开发停止。然而, 浏览器和它们共有的Web标准的实现得以保留了下来。

某些根本性的差异渐趋明朗。新Web技术的研究放慢了脚步。开发者们专注于创建清晰的HTML、使用优雅的CSS (Cascading Style Sheets, 层叠样式表) 和提升JavaScript技能。软件工具厂商实际上已经停止兜售他们承诺会令用户叫绝的新技术, 人们也无需永无停歇地学习新知识。转眼间, 我们都成了某一技术领域的专家, 成了确保股票市场不会崩盘的中坚力量。

这一轮“新Web”发展的焦点, 是在添加另外一层之前, 掌握最基本的Web技术。开发者们没有再次使用新技术构建旧应用, 而是致力于革新旧技术。同以前相比, 更多的非技术人群对使用Web应用程序乐在其中。而且, 在某些领域, 这些应用程序比类似的桌面应用程序表现得还要出色。也许是今天的Web应用程序具有相对轻便的计算机加载特性, 使得安装笨重的软件相形见绌。也许是设计师们对CSS充分地理解使得Web界面的设计和可用性具有了超前的征兆。无论如何, 新Web给人的感觉真好。

博客和社区网络等以人为本的应用程序带来了Web的繁荣——应用程序与Web完美地融为一体。用户能够通过类似文档的结构, 进行搜索和浏览。然而, 管理这些新型的应用程序并没有看起来那么简单。当用户登录到博客服务管理博客中的文章时, 由于文章遵循相同的文档浏览结构, 导致这一任务十分笨拙: 首先, 用户提交他们的凭证并等待账户页面加载, 然后, 选择想要管理的部分并等待该部分的页面加载……。这个过程涉及相继加载一系列文档, 因而导致了乏味而且不一致的用户体验。虽然相继加载页面是Web与生俱来的天性, 而且也能够完美地适用于许多任务, 但有效地管理博客则需要完全不同的方式。要让这种类似博客的

界面更具可用性，那么界面对用户而言就需要作为一个单独的响应主体存在。这是旧Web技术难以胜任的。

异步JavaScript技术的发展成为一个转折点，它消除了Web应用程序中令人讨厌的页面加载这一可用性问题。通过这种技术，JavaScript异步地向服务器发送数据请求，而且更新网页也不会导致刷新。以前不流畅的Web应用程序越来越像是完全一体化的桌面应用程序，操作中不会出现停顿或者整体性的界面刷新。随着JavaScript应用的日益增多，开发者们也把越来越多的编程逻辑放在了应用程序的客户端，进而减轻了服务器的负担：数据只在必要时加载，而且客户端应用程序也能保留以前在服务器上处理的状态信息。作为一种附加价值，可用性的改进也带来了性能的改进。而这种被称为Ajax（Asynchronous JavaScript and XML，异步JavaScript和XML）的，起初看起来有一点像hack的技术，已经得到Web用户和开发者的广泛认同。

使用Ajax技术构建Web应用程序，导致了Web应用程序架构的一次重大改变。今天的Web应用程序使用的界面不再通过Web服务器上的模板生成，而是在浏览器中通过JavaScript生成。应用程序的客户端本身变成了一个应用程序，而且在很大程度上与服务器保持了分离。开发者们只要对基本的Web技术掌握得足够好，就能构建完整的客户端应用程序——而且，某些情况下甚至不需要服务器——除非分发应用程序。虽然扩展Web技术的能力是件不太可能的事，但是，通过这些Web技术实现的功能却比任何一家公司开发的技术都要好。使用Ajax构建的应用程序看不到什么竞争：这些应用程序不断地被开发、分发和采用，比通过竞争性技术构建的任何应用程序的速度都要快，其中也包括那些使用传统的桌面应用程序开发工具构建的程序。

由于采取Ajax的应用程序越来越多，竞争性和应用复杂性也不断增加，使得Ajax应用程序的开发面临着可扩展性的考验。针对此类问题，其他应用程序开发技术是通过面向对象的设计和强大的开发工具来处理的。然而，从技术中发展而来的Ajax并不是用来构建应用程序的，更不必说构建大型应用程序。而且，也不存在类似辅助桌面应用程序开发的工具，来辅助JavaScript构建大型应用程序。这种支持是以框架的形式出现的，框架为构建应用程序提供了更好的基础（比如Dojo，就是一个社区努力的结晶）。但是，无论为构建Ajax应用程序开发的库有多少，Ajax的用途正日益与设计它的初衷背道而驰。

## 1.2 重新定义Web应用程序

通过一种解释型脚本语言——JavaScript，来编写Ajax应用程序的代码逻辑，使得开发者很难像使用桌面应用程序的开发语言一样，灵活地处理应用程序的规模和复杂性。同Ajax类似，桌面应用程序也有框架。然而，这些框架通常都会给应用程序带来更多的负担，并且往往也不能有效解决应用程序的扩展性问题。更重要的是，桌面应用程序有很多软件开发工具，这些工具比Ajax应用程序的开发工具要成熟得多。而且，这些软件开发工具成熟得也更快，因为它们也可以用于服务器端应用程序的开发。服务器端应用程序不仅更复杂，而且也需要更多的开发

工作量。

除了工具匮乏之外，JavaScript也缺少能够显著提高构建复杂应用程序能力的语言特性，包括可以肯定的面向对象编程的结构和在编译时捕获错误的能力。

很多公司都认识到了Ajax应用程序的成功和局限性，并参与到拿出Ajax的替代方案、提供更丰富的Web体验的行列中来。通过这些技术构建的应用程序（包括Ajax应用程序），都可以被归类为RIA（Rich Internet Application，富因特网应用程序）。

下面几节，我们将讨论取代Ajax技术构建RIA的一些方案。无论这些替代技术之间如何针锋相对，它们中间是否有一种技术能够具备足够的实力，最终成为RIA的事实标准，都有待于时间的检验。而且，Ajax很有可能继续保持其生命力。毕竟，Ajax在众多厂商实现的Web标准技术基础上，没有借助任何一家大公司声称出于善意的市场推广，自然而然地出现并被人们所接受。要放弃一种有效的技术，除非有一种方式能够解决它的问题，而且不会导致应用程序的用户安装由一家公司临时支持的另一种插件。

### 1.2.1 Adobe的Flash与Flex

Adobe Flash是有史以来最成功的浏览器插件——98%能上网的桌面电脑中都安装了该插件<sup>①</sup>。它的成功得益于相对较小因而容易下载的插件和插件播放的SWF文件、高效的动画性能以及便于图形设计者使用的构建SWF文件的开发工具。

Flash作为一款播放器已经广泛地应用到广告和游戏当中。而且，最近又用作以Adobe Flex构建的应用程序的播放器。Adobe Flex是构建基于Flash的应用程序的软件开发工具包，它使用MXML（Multimedia eXtensible Markup Language，多媒体可扩展标记语言）和ActionScript，在IDE（Integrated Development Environment，集成开发环境）中构建最终编译为SWF文件的应用程序。如果说Flash开发环境是为图形设计师量身打造的，那么Flex就是为软件开发人员定制开发的。而且，Flex还提供了与J2EE（Java 2 Enterprise Edition，Java 2企业版）服务器通信的远程和消息等数据服务。

虽然开发者通过Flash能够创建令人印象深刻的动画界面，但它基于绘画的机制可能会导致可用性問題，特别是当应用程序中使用的自定义控件与用户更熟悉的本地OS控件发生冲突时更为明显。Adobe最终可能会像Java处理SWT（Standard Widget Toolkit，标准部件工具包）一样来解决这个问题。

由于Flash只有一家公司推动，因而它不是基于标准的技术。为此，Adobe仍然需要为这一技术的市场占有率而竞争。这也使得Flash的前景不如Web技术那样可靠。比如，如果微软向一种竞争性产品中投入足够的资金，那么很可能会让Flash像以前的Netscape一样灰飞烟灭。

① 参见[www.adobe.com/products/player\\_census/flashplayer/](http://www.adobe.com/products/player_census/flashplayer/)。——译者注

### 1.2.2 微软的Silverlight

微软长期以来一直在支持基于Windows操作系统的客户端应用程序的开发。微软的工具和文档始终都是非常完善的。而且，他们也一贯致力于推进Web技术的发展。然而，微软过去的许多努力与Web面前人人平等的民主本性形成了鲜明反差。微软最初不喜欢Java Applet通过浏览器发布更高级的应用程序，于是决定从IE中删除JRE（Java Runtime Environment，Java运行时环境），并代之以嵌入ActiveX对象的能力。ActiveX对象使用的是微软专有的COM（Component Object Model，组件对象模型）技术。这个决定为浏览器实现引入了不确定性，使得有的浏览器支持Java，而有的却支持ActiveX。这种非标准化的尝试，正是导致ActiveX浏览器插件没有在今天普及的一个最重要原因。

微软从这次错误中吸取了教训，如今又基于.NET技术发布了作为许多浏览器插件的Silverlight（参见<http://silverlight.net/>）。通过Silverlight，开发者可以使用比JavaScript更先进的开发技术来构建RIA。然而，这个需要微软提供完善的.NET Framework工具和支持的解决方案，还没有得到广泛的分发。并且，Silverlight仍然由微软而不是标准组织所控制。

### 1.2.3 Java FX

Java由于其Java Applets拥有广泛的浏览器分发也曾有过一段惬意的时光。Java Applets通过内置于浏览器的Java运行时环境能够提供比HTML和JavaScript更高级的可编程功能。但是，同下载快、界面有冲击力的Flash相比，Java Applets也背负了下载缓慢、界面丑陋的名声。微软从IE中删除对Java Applets的支持，等于为它在浏览器中的生命划上了句号。

在Flash和Ajax成功地感召下，开发者们希望在浏览器中实现更多的可编程功能。而且，微软也带着Silverlight加入了游戏。这对于Java继续涉足基于浏览器的应用程序，似乎恰逢其时。

在微软发布Silverlight 1.1几天之后，Sun公司于2007年5月在JavaOne大会上发布了Java FX。Java FX家族技术有助于创建RIA。在本书出版的时候，Java FX由运行于标准JRE上的Script和Mobile模块组成，因此想要适当地评估这个方案还为时尚早。

## 1.3 基于Ajax的软件工程

Google以多少有些炫耀的方式发布了它们的Web工具箱。尽管同源自Google的其他新技术一样，GWT刚一发布就立即成为Digg.com和Slashdot.org的头条新闻<sup>①</sup>，但它还不是人们公认的一款伟大的Google产品。随之而来的评价中有赞誉也有轻蔑，不过，讨论在整体上还算平静。看起来，人们好像对这个工具箱在Web前景中的定位还心存疑虑，怎么说这都是一种开发Web

① 成为Digg.com的头条新闻一般被称为dugg，而成为Slashdot.org的头条新闻一般被称为slashdotted，这两个网站是以IT技术为主题的社区新闻网站。——译者注